Rapport TP2 Machine Learning

Description de la solution

Nous avons créé une solution qui vous permet d'importer, via une interface utilisateur front-end, une image d'animal à une API back-end. Cette dernière traite votre image en la classifiant selon sa base de données et vous retourne une prédiction sur l'animal.

Comment ce peut répondre à vos besoins

Notre solution peut être utilisée pour identifier des espèces animales, que ce soit dans des zoos, par des photographes ou à travers des caméras de surveillance.

Présentation des outils

**ML.NET** : C'est une extension de la plateforme .NET de Microsoft, créée en 2018, qui permet d'implémenter de l'apprentissage automatique dans vos applications .NET. Avec ML.NET, vous pouvez entraîner un modèle pour qu'il fournisse des prédictions basées sur les données d'entrée.

**React** : C'est une bibliothèque JavaScript développée par Facebook en 2013. Elle facilite la création d'applications web dynamiques, de manière plus simple qu'avec JavaScript seul. React est souvent utilisée pour développer des interfaces utilisateur dans le cadre du développement web.

Description de l’algorithme (Forces et Faiblesses)

Notre solution repose sur l'utilisation d'un algorithme d'apprentissage profond fourni par ML.NET, une bibliothèque d'apprentissage automatique robuste et polyvalente de Microsoft. Le cœur de l'algorithme est un modèle de classification d'image basé sur des réseaux de neurones convolutifs (CNN), spécialement optimisé pour l'interprétation et la reconnaissance visuelle d'images.

Forces :

- Haute précision : Grâce à sa capacité à apprendre des caractéristiques hiérarchiques des images, le modèle offre une grande précision dans la classification des différentes espèces animales.

- Automatisation : Une fois entraîné, le modèle effectue des prédictions sans intervention humaine, ce qui permet un traitement rapide et efficace des demandes d'analyse d'image.

- Adaptabilité : Le modèle peut être réentraîné avec de nouvelles données pour reconnaître de nouvelles catégories d'animaux, rendant la solution évolutive et adaptée aux besoins changeants.

Faiblesses :

- Dépendance aux données : La qualité et la quantité des données utilisées pour l'entraînement ont un impact direct sur les performances du modèle. Un ensemble de données biaisé ou trop petit peut limiter l'efficacité du modèle.

- Manque de transparence : Comme avec la plupart des modèles d'apprentissage profond, il peut être difficile de comprendre comment et pourquoi le modèle arrive à une prédiction spécifique (problématique de la 'boîte noire').

- Ressources de calcul : Le processus d'entraînement requiert des ressources de calcul substantielles, potentiellement limitantes pour les organisations avec des infrastructures informatiques moins développées.

Processus d’entrainement et conception de l’application

Nous avons commencé notre projet par la collecte d'un échantillon de plus de 900 images d’animaux triée en 16 catégories. Le modèle a été entraîné au sein d'une application console, utilisant notre échantillon d’images. Une fois l'entraînement achevé, le modèle a été intégré dans une API backend conçue pour gérer les requêtes venant de notre application cliente. Celle-ci a été spécifiquement développée pour permettre aux utilisateurs de soumettre des images et de recevoir les prédictions de notre modèle d'intelligence artificielle.

Évaluation des performances

Notre modèle s'est révélé assez bien performant lors des tests sur un échantillon de 900 images. Le modèle reflète une très bonne capacité à classer correctement diverses espèces animales. Cependant, les performances du modèle pourraient encore être améliorées en augmentant la taille de l'échantillon d'images.

Captures de la solution en action

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

Notre opinion sur les technologies expérimentées

**ML.NET : Nous avons été surpris par la facilité de compréhension de ML.NET, qui a dépassé nos attentes. Il offre une vaste gamme de modèles, bien au-delà de la simple reconnaissance d'images, pour l'apprentissage automatique. Avec de bonnes connaissances en .NET, vous pourrez facilement intégrer un modèle dans votre application.**

**React : Même s'il se révèle légèrement plus complexe à maîtriser que HTML/CSS/JavaScript, React offre une expérience de développement nettement plus agréable grâce à son architecture basée sur les composants. Cette approche modulaire non seulement simplifie la gestion de l'interface utilisateur mais encourage également la réutilisation du code, ce qui améliore l'efficacité du développement et la maintenance des applications.**

Captures du Kanban

**Semaine 1** :

**A screenshot of a computer

Description automatically generated**

**Semaine 2 :**

A screenshot of a computer

Description automatically generated

**Semaine 3 :**

A screenshot of a computer

Description automatically generated

**Semaine 4 :**

A screenshot of a computer

Description automatically generated